

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

Projekt dyplomowy

*Symulacyjny model adaptacyjnego tempomatu*

*A simulation model of adaptive cruise control system*

Autor: *Jakub Burczyk*

Kierunek studiów: Automatyka i Robotyka

Opiekun pracy: *dr inż., Marek Długosz*

Kraków, 2021

Spis treści

[1 Wstęp @TODO 3](#_Toc89112672)

[1.1 Charakterystyka problemu @TODO 3](#_Toc89112673)

[1.2 Cel oraz założenia pracy @TODO 3](#_Toc89112674)

[1.3 Zawartość pracy @TODO 3](#_Toc89112675)

[2 Środowisko symulacyjne – CARLA 3](#_Toc89112676)

[2.1 Czym jest CARLA 3](#_Toc89112677)

[2.2 Możliwości symulatora 3](#_Toc89112678)

[2.3 Interfejs symulatora 3](#_Toc89112679)

[3 @TODO 4](#_Toc89112680)

[3.1 @TODO 4](#_Toc89112681)

[4 Bibliografia 4](#_Toc89112682)

# Wstęp @TODO

## Charakterystyka problemu @TODO

@TODO

## Cel oraz założenia pracy @TODO

@TODO

## Zawartość pracy @TODO

@TODO

# Środowisko symulacyjne – CARLA

## Czym jest CARLA

Symulator CARLA jest środowiskiem symulacyjnym zaprojektowanym w celach badawczych i weryfikacyjnych systemów pojazdów autonomicznych. Jedym z celów przyświecających twórcom była całkowita transparentność dla użytkownika i mozliwość rozwoju projektu przez członków społeczności. Dostępny jest on w formie open-source pod licencją MIT[[1]](#footnote-1) zarówno na komputery z systemem Windows i Linux. Został zbudowany na bazie silnika Unreal Engine 4, który umożliwia wierną symulację oświetlenia i fizyki obiektów. Wykorzystuje on standard OpenDRIVE, za pomocą którego definiowane są sieci dróg oraz ich parametry związane ze sterowaniem ruchem ulicznym.

## Możliwości symulatora

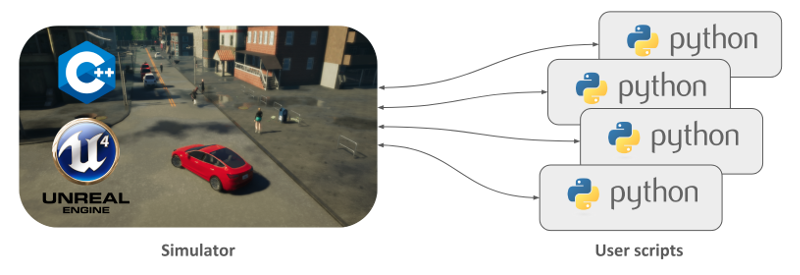
Dzięki zastosowaniu nowoczesnych rozwiązań, wsparciu gigantów technologicznych takich jak Intel, Samsung oraz producentów branży automotive takich jak Mercedes, Toyota czy Valeo symulator zaopatrzono w szereg zaawansowanych funkcjonalności.

Składają się na nie między innymi:

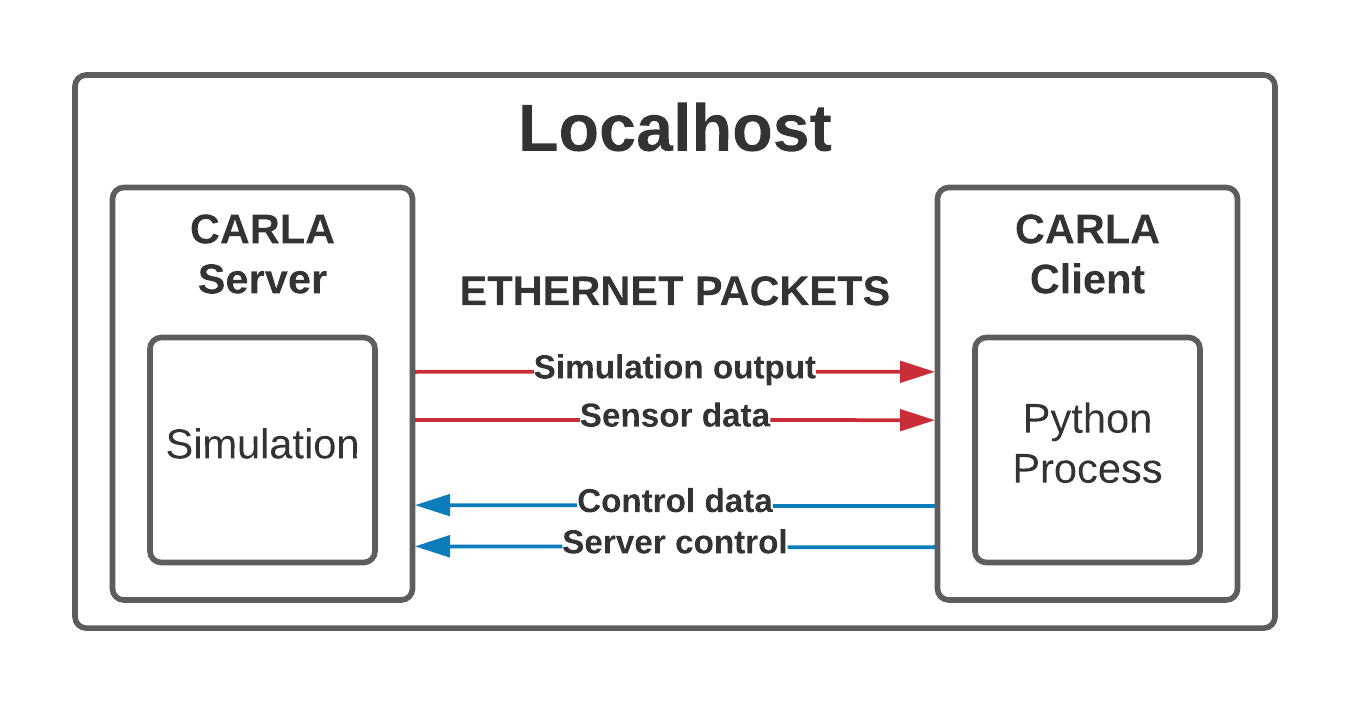
* Symulacja fizyki pojazdów, charakterystyk trakcji, oddziałujących sił i oporów

## Interfejs symulatora

Symulator można podzielić na dwie części, serwerową i poszczególnych klientów. Część serwerowa odpowiedzialna jest między innymi za symulację sensorów, obliczenia fizyki obiektów, aktualizację świata i aktorów[[2]](#footnote-2). Klienci natomiast, za pośrednictwem API[[3]](#footnote-3) w języku Python, mogą komunikować się z serwerem poprzez wysyłanie komend i meta-komend zawierających informacje np. o sterowaniu lub zmianie parametrów symulacji. Obie części symulatora mogą działać równolegle na jednej maszynie, do której odwołujemy się poprzez hosta lokalnego, bądź poprzez sieć komputerową. Umożliwia to uruchomienie serwera na jednostce o dużej mocy obliczeniowej, która udostępnia klientom wysokiej jakości symulację, odciążając tym samym zasoby klientów, które mogą być przeznaczone na algorytmy sterowania.



Rysunek 2.1 Ogólny schemat działania symulatora  
źródło: https://carla.readthedocs.io/en/0.9.12/start\_introduction/



Rysunek 2.2 Schemat komunikacji w obrębie maszyny lokalnej

Należy zauważyć, że struktura ta może być rozszerzona o dowolną ilość klientów, z czego żaden nie jest limitowany co do ilości i rodzaju symulowanych obiektów.

# @TODO

## @TODO

# Bibliografia

1. A. Dosovitskiy, G.Ros, F. Codevilla, A. López, V. Koltun – „CARLA: An Open Urban Driving Simulator”; PMLR 78:1-16
2. Dokumentacja symulatora CARLA (w wersji 0.9.12) dostępna pod adresem: https://carla.readthedocs.io/en/0.9.12/

1. Licencja MIT – rodzaj licencji, który pozwala na dowolne modyfikacje i dystrybucję danego oprogramowania. [↑](#footnote-ref-1)
2. Aktor – w rozumieniu symulatora CARLA, jest instancją obiektu symulacji, może być to między innymi pojazd, sensor, obserwator czy obiekty związane ze sterowaniem ruchem drogowym [↑](#footnote-ref-2)
3. API – Application Programming Interface [↑](#footnote-ref-3)